Labs – Sprint 2 – Loom Video Notes

**Objective**: Integrate the frontend application with an API by updating the GraphWrapper.jsx file and data visualizations to use the API instead of the test\_data.json file. After integration, the three graphs on the Graphs page should display data from the API. The specific API endpoints utilized are the Fiscal Year Data (/fiscalSummary) and Citizenship Data (/citizenshipSummary).

**Implementation**:

1. **Enhancement of React Component**:
   * Integrated an Axios call within the updateStateWithNewData function.
   * Utilized async/await and a try/catch block to handle the API requests and ensure proper data display.
2. **Detailed Code Walkthrough**:

**updateStateWithNewData Function**:

* + **Purpose**: Fetch data from the server based on specified parameters (years, view, office) and update the application state using a callback function.
  + **Function Body**:
    1. **Initialization**:
       - Initialized three variables to store the results of API calls: fiscalSummaryResult, heatMapResult, and citizenshipSummaryResult.
    2. **API Calls Based on View**:
       - **Time-Series View**:
         * GET request to fetch fiscal summary data for the specified years and office.
         * State updated with fetched data.
       - **Office Heat Map View**:
         * GET request to fetch fiscal summary data for the specified years and office.
         * State updated with fetched data.
       - **Citizenship View**:
         * GET request to fetch citizenship summary data for the specified years and office.
         * Additionally, fetches fiscal summary data.
         * Combines the results and updates the state with fetched data.
    3. **Error Handling**:
       - Logs any API call errors to the console.